Chapter 11
A Distributed Requirements Collaboration Process

Brandon Rydell
Portland State University, USA

Sean D. Eby
Portland State University, USA

Carl Seaton
Portland State University, USA

ABSTRACT
Project teams struggle to accurately document the negotiations and trade-offs required to lead teams to concise, prioritized requirements. The main reasons for this challenge are (1) a lack of objective criteria to evaluate requirements, (2) a lack of freely available tools to channel negotiations into objectively characterized requirements which can be comparatively evaluated, and (3) a failure to capture the rationale for these decisions.

While there are numerous publications and tools addressing the challenges of software requirements engineering, many organizations today still manage requirements as tabular lists. Individual requirements are often modified during negotiations between groups such as engineering, marketing, and management. Frequently, much of this negotiation takes place informally in numerous face to face meetings where much of the rationale for changes and trade-offs are not documented, and are subsequently lost. This challenge becomes increasingly difficult with physically and / or temporally distributed teams, when not all stakeholders are present during these meetings. This break in communications leads to costly rework and lost opportunities as information gets delayed, misinterpreted, or even lost entirely.

This chapter outlines a new requirements collaboration process to address these issues. This new process is based on work done by Karl Wiegers in his paper Prioritizing Requirements (Wiegers, 1999), and is extended to include a method of proposing alternate requirements, documenting the negotiation of priority among interested parties, and a way to rationally select priority requirements based on an

objective measure of their relative merit. The chapter also introduces an application prototype called the Distributed Requirements Collaboration Tool (DRCT) that was built for use by distributed teams to support the requirements negotiation process described above and also to address the issue of capturing rationale as requirements are negotiated. Lastly, the experience using the DRCT is discussed in order to identify priority requirements for a more functional version of the DRCT.

INTRODUCTION

Fred Brooks said:

The hardest single part of building a software system is deciding precisely what to build. No other part of the conceptual work is as difficult as establishing the detailed technical requirements, including all the interfaces to people, to machines, and to other software systems. No other part of the work so cripples the resulting system if done wrong. No other part is more difficult to rectify later. (Brooks, 1986, p. 11.)

Requirements elicitation, specification, and management are the fundamental tools used by software engineers when deciding precisely what to build. A common challenge with requirements management for organizations is to accurately document the negotiations and trade-offs that are required to lead teams to concise prioritized requirements, especially within distributed teams.

Some reasons for this challenge include:

1. A lack of objective criteria to evaluate requirements (frequently requirements are dictated by vertical teams like marketing / engineering, and presented as simple lists).
2. A lack of freely available tools to channel negotiations into objectively characterized requirements which can be comparatively evaluated.
3. A failure to capture the rationale for decisions, including the explicit trade-offs negotiated and the alternatives considered.

While there are countless books, papers and websites addressing the challenges of software requirements engineering, and there are even commercial requirements management tools available, many organizations today still use basic documents or spreadsheets to manage simple requirements lists. This is due to a lack of software requirements engineering knowledge, skill and application.

In some more advanced organizations, requirements are often modified during negotiations between engineering (technical feasibility and cost), marketing (market benefit), and management (schedule, resources and strategic business goals). Frequently much of this negotiation takes place informally face to face during meetings where much of the rationale for alternatives, changes and trade-offs are not recorded. This challenge becomes increasingly difficult with physically and / or temporally distributed teams.

While these informal negotiations are useful in creating a good product, their success rests on the assumption that all stakeholders are present and will remain on the project indefinitely. When the stakeholders are separated by time, turnover, or geography, this assumption breaks down. In particular, unless distributed teams have a way to participate in the discussions and be informed of the decisions, costly mistakes will occur during the product’s development. These errors will compound in future releases as time and potentially different personnel increase information loss.

In this chapter we outline a new requirements collaboration process which addresses these issues and as a result offers a process that is more suited to the needs of distributed teams. The development of this new process began by using Karl Wiegers’ requirements prioritization process as